

Team he-hu-ho-jo Conceptual Sketch

Key Purposes and Social Needs

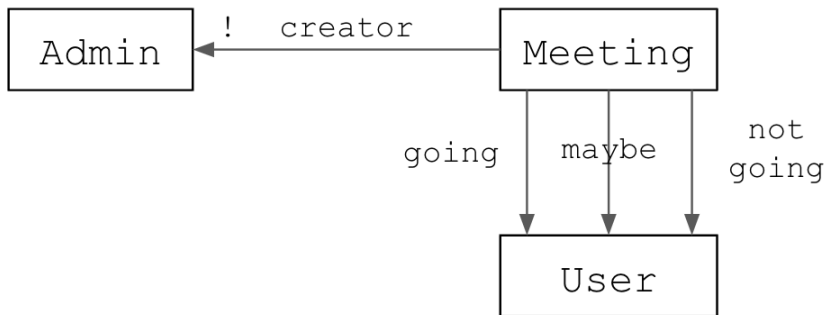
Our project aims to tackle the key social problem of low civic engagement in one's local community. The media places a heavy emphasis on elections and measures at the state and/or national level. However, there are highly critical actions happening at the municipal level every single day that are often forgotten about since they are not heavily reported. This undermines the significance of local meetings. Oftentimes, decisions that directly impact community members are being made at these meetings. Our project seeks to both inform users of local meetings that may pertain to their particular political/social interests, as well as incentivize users to become more involved within their local community by using gamification to reward users who attend local meetings.

Key Concepts

Name: **Meeting**

Purpose: to gather members of the community to discuss topics relating to the city

State:



going, maybe, not going have no intersections
(for each meeting, a user is either going,
maybe, or not going (or none))

Actions:

- `addGoing(id: Int, u: User) {`
 `let meeting = Meetings.locate(id);`
 `meeting.going.add(u);`
- `addMaybe(id: Int, u: User) {`
 `let meeting = Meetings.locate(id);`
 `meeting.maybe.add(u);`
- `addNotGoing(id: Int, u: User) {`
 `let meeting = Meetings.locate(id);`
 `meeting.notGoing.add(u);`

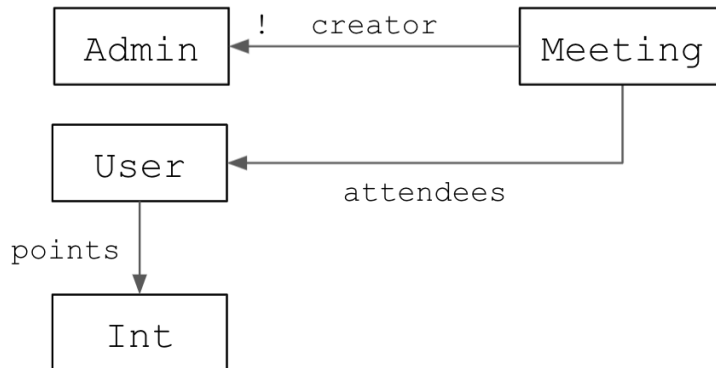
Operational Principle:

All `u: User` | if (`addGoing(id, u)` or `addMaybe(id, u)`) and not `addNotGoing(id, u)`, then corresponding meeting will appear in user's 'Upcoming Meetings' feed

Name: **Admin Privilege**

Purpose: provide authorization access to manage meetings

State:



Actions:

- createMeeting(name: String, host: Admin, time: Nat, location: String, agenda: String, tags: List(String)):
 m = new Meeting;
 m.name = name;
 m.time = time;
 m.location = location;
 m.agenda = agenda;
 m.tags = tags;
 m.host = host;
 m.id = count; // count will be a unique integer stored globally
 m.attendees = [];
 m.going = [];
 m.maybe = [];
 m.notGoing = [];
 DB.add(m);
- deleteMeeting(id: Int, host: Admin):
 let meeting = Meetings.locate(id);
 if (host.name===meeting.host) {
 DB.remove(meeting);
 }
- editMeeting(id: Int, host: Admin, newValue: *new value type for respective field*, field: String) {
 let meeting = Meetings.locate(id);
 if (host.name===meeting.host) {
 meeting.{field} = newValue;
 }
- addAttendance(id: Int, attendees: List(User)) {
 let meeting = Meetings.locate(id);
 meeting.attendees = attendees;

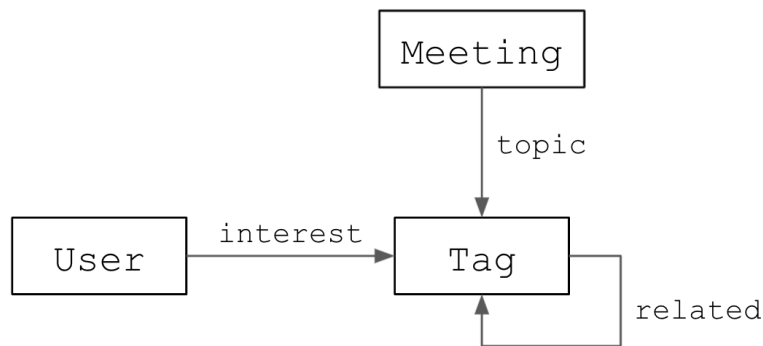
Operational Principle:

if admin a creates meeting m, then m will appear in user u's "all meeting" feed
if admin adds addAttendance(i, attendees), each of the attendees earns a point

Name: **Recommend Topics**

Purpose: More easily discover meetings of probable interest

State:



Actions:

- likeTopic(t:Tag, u:User):
 - $u.\text{interest} += t$
- unlikeTopic(t:Tag, u:User):
 - $u.\text{interest} -= t$
- getRecommendedTags(t:Tag, u:User):
 - $u.\text{interest}$

Operational Principle:

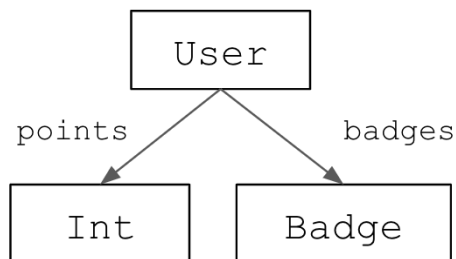
for user u, if u likes tag t, then t's related set of tags R will be recommended to u as a suggested topic.

if u likes tag r in R, then meetings with tags including r will be included in the user's meetings with probable interests.

Name: **Incentive System**

Purpose: introduce game mechanics to web app to to increase user participation

State:



Actions:

- givePoints(u: User, p: Point)
 - $u.\text{points} += p$

- giveBadge(u: User, t: Threshold, b: Badge)
 if u.points > t
 User.badges += b
- getPoints(u: User, p: Point)
 return u.points
- getBadges(u: User, b: Badge)
 return u.badges

Operational Principle:

if user demonstrates engagement, admin award point to user
if user is awarded points beyond a threshold, user is given a badge

Questions

1. Committed Unity goes beyond CRUD in several ways. We are introducing functionalities beyond just allowing admins to post meetings, and users to see these meetings. One such example of this is the gamification system. We chose to implement the not-widely-used concept of gamification to actually incentivize users to go to meetings. Otherwise, our application does not proactively try to increase civic engagement. For our gamification system, the user will receive a 'point' for attending a meeting, and if they receive enough points, they will achieve a distinguished status which will make them eligible for benefits (these benefits are to be determined by the municipality or city that is using this app). This will require operations beyond CRUD, including storing the points of each user, as well as tracking which users attend which meetings. Our application also goes beyond CRUD by curating specific meetings to users based on that particular user's interests, rather than just displaying all upcoming meetings to a user (we will have an option to view all upcoming meetings, but in addition we will suggest meetings to users). Similarly, our app goes beyond CRUD by offering tags to users that are similar to their particular interests. This allows them to learn about tags that they may have not been aware of, so they can broaden their interests and thus learn about meetings that they may have not been interested in otherwise. These two concepts will require intricate matching processes that take a meeting's tags, as well as a user's tags, into consideration to accurately and robustly recommend meetings and tags to users based on their particular interests, which goes beyond simply creating, reading, updating, and deleting items.
2. These concepts come with several challenges. For our gamification system, this will potentially be challenging since we need to robustly track users' attendance at meetings for rewarding points (which will become difficult if we try to scale this app up), as well as figure out what constitutes the aforementioned distinguished status that makes a user eligible for benefits (i.e. does obtaining 30 points give a user this status? 60? Should it depend on the size of the municipality and the number of meetings they host?). Another challenge will be balancing the suggestion of meetings of interest to users with bipartisanship. There is an inherent ethical dilemma in suggesting users content/meetings that solely relate to their political interests, as this can spiral into an echo chamber that leads to increased political polarization. We must carefully choose in our design process how to suggest meetings to users in a way that does not divide populations and increase partisanship.

3. One last design area that will likely be challenging is determining meeting tags in a robust way, should we scale up our app's scope in the future. Ideally, people should be able to create meeting tags, since community meetings can touch upon very niche and specific topics, and our bank of tags should be able to grow as more users use the app. However, if we keep our bank of tags dynamic, this could lead to downstream effects (i.e. capitalization/spelling/symbol errors) that make it difficult to suggest meetings to users based on their specific interests, e.g. a meeting is tagged "Water and Power" but a user's interest is "Water & Power."